```
module decoder (T0,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15, dec_in);

output T0,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15;
input [3:0] dec_in;
reg T0,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15;

always @(*)
begin

case(dec_in)
4'b0000:
begin
T0=1;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b0001:
begin
T0=0;T1=1;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b0010:
begin
T0=0;T1=0;T2=1;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b0011:
begin
T0=0;T1=0;T2=0;T3=1;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b0100:
begin
T0=0;T1=0;T2=0;T3=0;T4=1;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b0101:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=1;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b0110:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=1;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
```

```
end

4'b0111:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=1;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b1000:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=1;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b1001:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=1;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b1010:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=1;T11=0;T12=0;T13=0;T14=0;T15
=0;
end

4'b1011:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=1;T12=0;T13=0;T14=0;T15
=0;
end

4'b1100:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=1;T13=0;T14=0;T15
=0;
end

4'b1101:
 begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=1;T14=0;T15
=0;
end

4'b1110:
begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=1;T15
=0;
end
```

```
default: begin
T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;T11=0;T12=0;T13=0;T14=0;T15
=1;
end
endcase
end

endmodule

module sequencer (clk,CLR,co_out);
input clk,CLR;
output [3:0] co_out;

reg [3:0] co_out;
wire  T0,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15;

initial
begin
co_out =4'bxxxx;
end

always @ (posedge clk)
begin
if (CLR==1 || co_out==4'b1111) co_out=4'b0000;
else co_out= co_out + 4'b0001;
end
decoder D1 (T0,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15,co_out);

endmodule



module sequencer_tb
(clk,CLR,co_out,T0,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15);

output clk,CLR;
output [3:0] co_out;
input T0,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15;

reg clk,CLR;
wire [3:0] co_out;

initial
begin
$dumpfile ("sequencer_tb.vcd");
$dumpvars;
clk=1'b0;
CLR=1'b0;
#250 $finish;
end
```

```
initial
begin
#3 CLR=1'b1;
#5 CLR=1'b0;
end

initial
begin
#53 CLR=1'b1;
#5 CLR=1'b0;
end

always
begin
#5 clk = ~clk;
end

sequencer C1 (clk,CLR,co_out);

endmodule
```